

CLAIMS

1. A computer network comprising:

a plurality of processors connected to said network, each of said processors comprising a plurality of first processing units having the same instruction set architecture and a second processing unit for controlling said first processing units, said first processing units being operable to process software cells transmitted over said network, each of said software cells comprising a program compatible with said instruction set architecture, data associated with said program and an identification number uniquely identifying said software cell among all of said software cells transmitted over said network.

2. The network of claim 1, wherein said second processing unit controls said first processing units by determining the programs of said software cells processed by said first processing units.

3. The network of claim 2, wherein each said first processing unit includes a local memory exclusively associated with said first processing unit and said first processing unit processes said programs from said local memory.

4. The network of claim 1, wherein each said processor further includes a main memory, said main memory including a plurality of banks, each said bank including a plurality of blocks, each said block being the lowest addressable unit of said main memory and having an associated

memory space in said main memory for storing information regarding the status of data stored in said block, an identification for a first processing unit and an address of a local memory associated with said first processing unit.

5. The network of claim 4, wherein said first processing units comprise means for using said associated memory spaces to synchronize said first processing units' reading of data from, and writing of data to, said blocks.

6. The network of claim 1, wherein each of said processors further comprises a direct memory access controller.

7. The network of claim 4, wherein each said first processing unit is operable to issue a synchronize read command to read data from said main memory to a local memory associated with said first processing unit and to issue a synchronize write command to write data from said local memory to said main memory.

8. A computer readable medium for storing a software cell for transmission over a computer network, said computer network comprising a plurality of processors, said software cell comprising:

a program for processing by one or more of said processors;

data associated with said program; and

a global identification uniquely identifying said software cell among all software cells transmitted over said network.

9. The computer readable medium of claim 8, wherein said software cell further comprises information for routing said software cell over said network.

10. The computer readable medium of claim 9, wherein said information includes an identification for one of said plurality of processors, said one processor being the processor to which said software cell is to be transmitted for processing.

11. The computer readable medium of claim 10, wherein said identification includes an internet protocol address.

12. The computer readable medium of claim 9, wherein said information includes an identification for one of said plurality of processors, said one processor being the processor from which said software cell originates.

13. The computer readable medium of claim 9, wherein said information includes an identification for one of said plurality of processors, said one processor being the processor to which information regarding the processing of said software cell is to be transmitted.

14. The computer readable medium of claim 8, wherein said software cell further comprises information

providing a plurality of direct memory access commands for one of said processors.

15. The computer readable medium of claim 14, wherein said information comprises a virtual identification for said one processor and addresses of a memory associated with said one processor for implementing said direct memory access commands.

16. The computer readable medium of claim 8, wherein said global identification is based upon the identity of one of said processors, said one processor being a processor creating said software cell, and upon the time and date of said creating.

17. The computer readable medium of claim 8, wherein said global identification is based upon the identity of one of said processors, said one processor being a processor transmitting said software cell, and upon the time and date of said transmitting.

18. A data stream of software cells for transmission over a computer network, said computer network comprising a plurality of processors, each of said software cells comprising:

a program for processing by one or more of said processors;

data associated with said program; and

a global identification uniquely identifying said software cell among all software cells transmitted over said network.

19. The data stream of claim 18, wherein each said software cell further comprises information for routing said software cell over said network.

20. The data stream of claim 19, wherein said information includes an identification for one of said plurality of processors, said one processor being the processor to which said software cell is to be transmitted for processing.

21. The data stream of claim 20, wherein said identification includes an internet protocol address.

22. The data stream of claim 19, wherein said information includes an identification for one of said plurality of processors, said one processor being the processor from which said software cell originates.

23. The data stream of claim 19, wherein said information includes an identification for one of said plurality of processors, said one processor being the processor to which information regarding the processing of said software cell is to be transmitted.

24. The data stream of claim 18, wherein each said software cell further comprises information providing a plurality of direct memory access commands for one of said processors.

25. The data stream of claim 24, wherein said information comprises a virtual identification for said one processor and addresses of a memory associated with said one processor for implementing said direct memory access commands.

26. The data stream of claim 18, wherein said global identification is based upon the identity of one of said processors, said one processor being a processor creating said software cell, and upon the time and date of said creating.

27. The data stream of claim 18, wherein said global identification is based upon the identity of one of said processors, said one processor being a processor transmitting said software cell, and upon the time and date of said transmitting.

28. A method for processing programs and data associated with said programs on a computer processor, said computer processor comprising a main memory, a memory controller, a plurality of first processing units, each said first processing unit including a local memory exclusively associated with said first processing unit, and a second processing unit, said method comprising:

storing in said main memory said programs and said data associated with said programs;

directing with said second processing unit any one of said first processing units to process one of said programs;

directing with said second processing unit said memory controller to transfer said one program and data associated with said one program from said main memory to the local memory exclusively associated with said one first processing unit;

instructing with said second processing unit said one first processing unit to initiate processing of said one program from said one first processing unit's local memory; and

in response to said instructing, processing with said one first processing unit said one program and said data associated with said one program from said local memory exclusively associated with said one first processing unit.

29. The method of claim 28, wherein said main memory is a dynamic random access memory.

30. The method of claim 28, wherein said main memory includes a plurality of memory locations, each said memory location including a memory segment exclusively associated with said memory location.

31. The method of claim 30, further comprising storing in each said memory segment status information indicating the status of data stored in said memory segment's associated memory location, the identity of a first processing unit and a memory address.

32. The method of claim 31, wherein said status information indicates the validity of said data stored in said

memory segment's associated memory location, said identity indicates the identity of a particular one of said first processing units and said memory address indicates a storage location within the local memory exclusively associated with said particular one first processing unit.

33. The method of claim 28, wherein each of said first processing units is a single instruction multiple data processor.

34. The method of claim 28, wherein each of said first processing units includes a set of registers, a plurality of floating points units, and one or more buses connecting said set of registers to said plurality of floating point units.

35. The method of claim 34, wherein each of said first processing units further includes a plurality of integer units and one or more buses connecting said plurality of integer units to said set of registers.

36. The method of claim 28, wherein said computer processor comprises an optical interface and an optical waveguide connected to said optical interface, and further comprising converting electrical signals generated by said processor to optical signals for transmission from said computer processor over said waveguide and converting optical signals transmitted to said processor over said waveguide to electrical signals.

37. The method of claim 28, wherein each said local memory is a static random access memory.

38. The method of claim 28, wherein said computer processor further comprises a rendering engine, a frame buffer and a display controller, and further comprising generating pixel data with said rendering engine, temporarily storing said pixel data in said frame buffer and converting with said display controller said pixel data to a video signal.

39. The method of claim 28, wherein the data associated with said one program includes a stack frame.

40. The method of claim 28, further comprising, during said processing of said one program and said data associated with said one program, transferring with said memory controller, in response to an instruction to said memory controller from said one first processing unit, further data from said main memory to the local memory exclusively associated with said one first processing unit and thereafter processing said further data with said one first processing unit from said local memory exclusively associated with said one first processing unit.

41. The method of claim 28, wherein said main memory comprises a plurality of memory bank controllers and a cross-bar switch for providing a connection between each of said first processing units and said main memory.

42. The method of claim 28, further comprising prohibiting with said memory controller each said first

processing unit from reading data from, or writing data to, any of said local memories with which said first processing unit is not exclusively associated.

43. The method of claim 28, further comprising, following said processing of said one program and said data associated with said one program, transferring with said memory controller to said main memory, in response to an instruction to said memory controller from said one first processing unit, processed data resulting from said processing of said one program and said data associated with said one program.

44. The method of claim 28, wherein said memory controller is a direct memory access controller.

45. The method of claim 28, wherein said computer processor is connected to a network and said one program is included within a software cell, said software cell containing a global identification uniquely identifying said software cell among all software cells transmitted over said network.